Chapter 3
The Installer in Detail

The choices provided in the installer can be confusing at first. After you've done a few installs, you get the hang of things, but when you're starting out, it's good to have a place where each choice, and the ramifications of those choices, is spelled out explicitly. Here's the place.

Since you're a programmer, you're probably itching to see what's going on under the hood. In the last chapter, we blew through the selections in the Installer fairly quickly. You probably are wondering "What about those other choices in the Installer?" and don't want to thumb all the way through to Chapter 33 to find out. Now it's time to discuss the other choices in the Installer and what the ramifications of those choices are.

Access Control Lists (ACL) Concepts

While one of the fundamental intents of a wiki is community editing of the pages, this doesn't necessarily mean that just any ol' traveler who wanders by can be considered part of the community. Particularly for DokuWiki, which was built for software documentation, it's important to be able to restrict who the community consists of, and what rights those members have.

As a result, DokuWiki uses a mechanism called Access Control Lists to allow the administrator to restrict access to the wiki as needed. This restriction may be as light as simply requiring a visitor to register before gaining access, or as sophisticated as organizing users into groups and assigning different rights (read-only, edit pages, create pages, etc.) to different areas of the wiki to each group.

The implementation of ACLs in DokuWiki is a two-step process. First, you choose whether to use ACLs or not. If you choose not, the game is over - you've got a wide open wiki with absolutely no security or restrictions - not even an administrator. If you do enable ACLs, you then have a second step to take - choosing how to structure the wiki - from wide open to completely closed or somewhere in between. I'll cover each of these permutations in the rest of this chapter.

Storage of installation and configuration settings

Installation and configuration settings are stored in a set of files in the /conf folder. I'll cover these settings in detail in Chapter 7, but here's the nickel tour so you can see how the various ACL options are stored as settings.

The 'acl.auth.php' file contains the Access Control Lists themselves. Except for the most extreme circumstances, you'll not edit this file yourself; instead, you'll use a tool called the ACL Manager (covered in Chapter 12) to do so.

The 'users.auth.php' file contains the user set up to access the wiki. For example, once you create an administrator or a regular user, this information is stored in users.auth.php. Again, except for the most extreme circumstances, you'll not edit this file directly; instead, you'll use the User Manager (also covered in Chapter 12).

Finally, the 'local.php' file contains configuration settings. You've already run across three of these - the default language, the name of the wiki and whether or not ACL is enabled. The configuration settings are covered in greater detail in Chapter 7.

Option 1: Enable ACL checkbox deselected

Our first option is to deselect the Enable ACL checkbox. If you do so, you've created a wide-open wiki with no users, no restrictions, no nothing. Totally open to anyone who wanders by. All the rest of the controls on the Install page disappear, as shown in **Figure 1**.

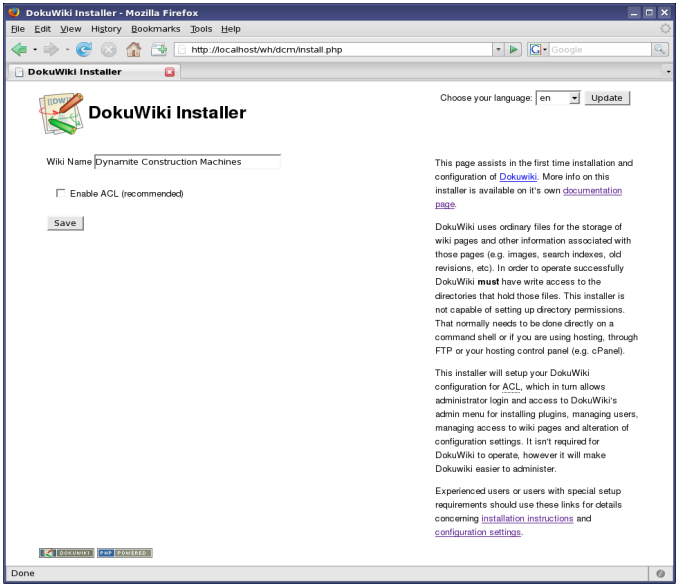

***Figure 1**. Enable ACL unchecked*

After you click 'Save' and continue on, you'll get a frighteningly stark page that allows creation and editing of pages willy-nilly, as shown in **Figure 2**.

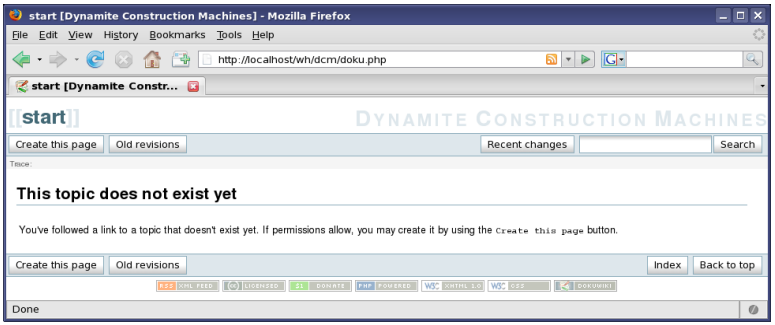

***Figure 2**. The original start page without ACL enabled.*

Notice that there is no "Login" button on this page as there was in Figure 16 of Chapter 2. Additionally, there is no way to enter the administration interface, which means that all configuration must be done manually by editing the DokuWiki configuration files in a text editor. If you don't need any access restrictions, you should still enable ACL and choose the "Open Wiki" option in the installer, which is covered in the next section.

Easy to use, not-so-easy to administer. But since we're working on private software documentation, not for us. The next three choices all feature the "Enable ACL" checkbox selected and one of the three options in the "Initial ACL policy" combo selected.

Under the hood: If you look in the 'conf' folder after deselecting ACL, you won't see acl.auth.php or users.auth.php files, and in local.php, there will be two settings - one for the default language and a second for the name of the wiki.

```
$conf['title'] = 'Dynamite Construction Machines';
$conf['lang'] = 'en';
```

There won't be a setting having to do with ACL. The absence of the setting means DokuWiki doesn't even know about ACL.

Option 2: Enable ACL, policy: open wiki

This option creates a wiki that is completely open, but an administrator account created and has user accounts enabled but not required. Usually, the only user account created is a super user who performs administration tasks like configuration. See **Figure 3**.

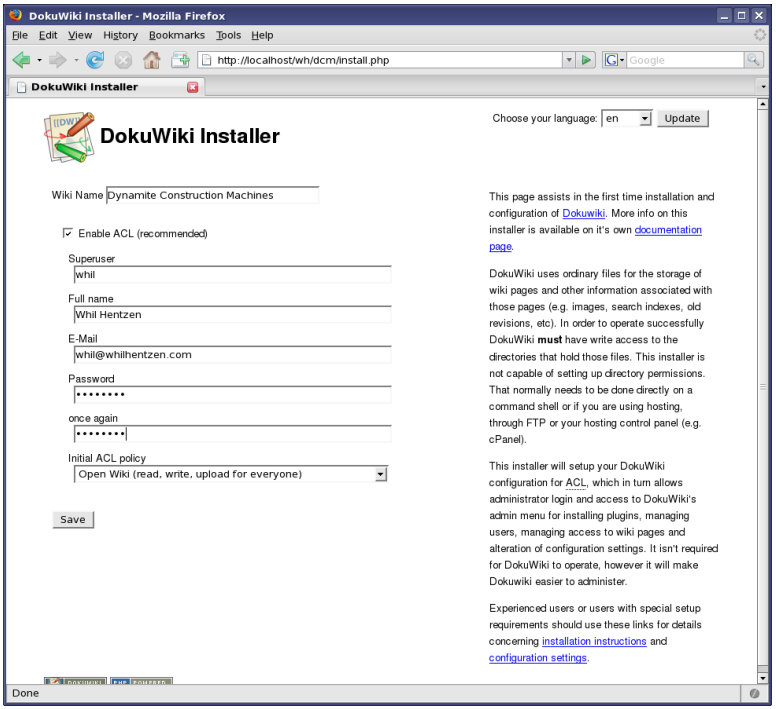

***Figure 3**. ACL enabled but wiki completely open.*

We've already seen what this permutation looks like. Figure 13 in Chapter 2 shows the Installer page with Enable ACL checked (yes, it's basically the same as Figure 3 here) while Figure 15 shows the "You're ready!" page. Once you click the "Continue to your new wiki" link (on the "You're ready!" page), you'll be greeted by **Figure 4**, that, much like Figure 3, is identical to Figure 16 in Chapter 2.

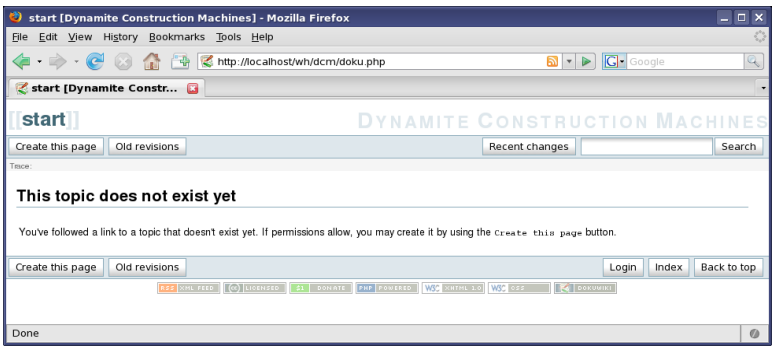

***Figure 4**. The start page of your brand new wiki with a Login button.*

Under the hood, local.php now contains two new entries.

```
$conf['useacl'] = 1;
$conf['superuser'] = '@admin';
```

The first option indicates that ACL is enabled, while the second assigns superuser capabilities to anyone who is a member of the 'admin' group.

Additionally, the acl.auth.php and users.auth.php files are created. (Finally!) The contents of the acl.auth.php file looks like this:

```
*               @ALL          8
```

This string means that all members of the group "ALL" (which is everyone) can read, create and edit pages, and that there will be a login button to allow a superuser to perform administrative tasks. The contents of this file will change in the next two configurations.

The users.auth.php file contains a single line like this:

```
whil:062e8c8d742d927sf7:Whil Hentzen:whil@whilhentzen.com:admin,user
```

This file contains the credentials of all users who have accounts; right now, only one user has an account - the super user who did the original install.

At this point, the wiki is completely open for editing by anyone who happens by. The only reason one would need to log in is to perform administration tasks. Administration includes tasks like setting configuration options, managing templates, installing plugins, and adding additional users.

User accounts, even in an open wiki, are useful for attribution - so people can see who contributes useful material to the wiki - and so that users can receive notifications of when pages are changed (called page subscriptions.)

Option 3: Enable ACL, public wiki

The second choice in the Initial ACL policy combo is "Public wiki", which, as it explains, allows anyone to read the contents of the wiki, but only logged in users to make changes. See **Figure 5**.

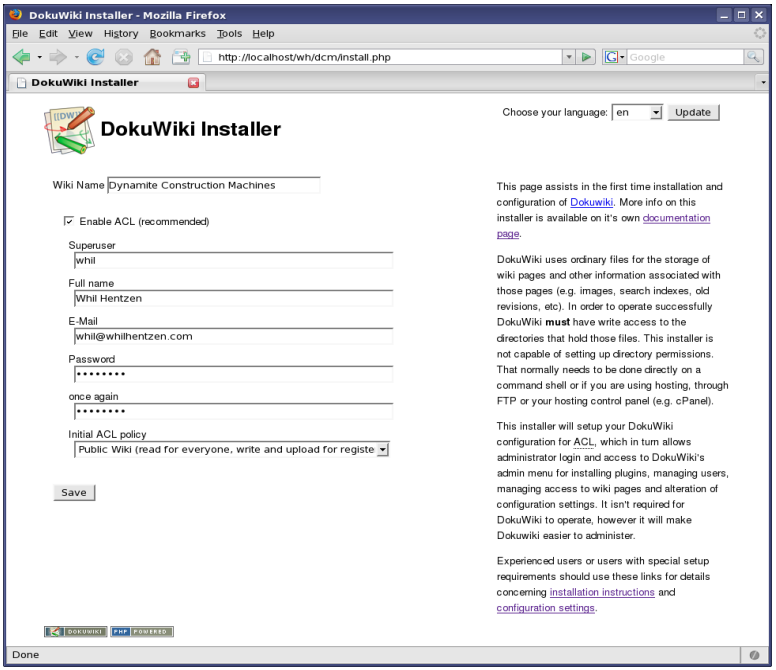

***Figure 5**. Setting the Initial ACL policy to 'Public wiki' in the DokuWiki Installer.*

The initial DokuWiki page is shown in **Figure 6**.

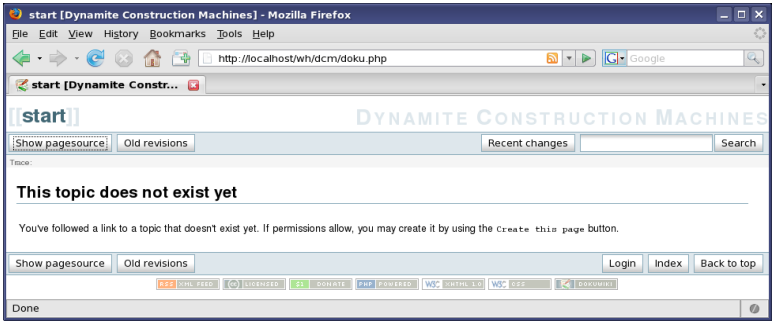

***Figure 6**. The start page of a new "Public" wiki.*

As you can see, it doesn't look a lot different than the Open wiki page shown in Figure 4. The one change you might notice is the button on the far left says "Show pagesource" instead of "Create this page". Additionally, there are some changes under the hood.

The contents of the acl.auth.php file have changed in this configuration.

```
*               @ALL            1
*               @user           8
```

These settings say that all can read (1), but you must log in to create/edit pages (8).

The contents of the other two /conf files haven't changed. 'local.php' still contains the same contents as for an Open wiki and the 'users.auth.php' file is also the same as for an Open wiki - a single entry for the original super user defined in the Installer screen.

Option 4: Enable ACL, closed wiki

Here's where it gets interesting. Selecting the third choice in the Initial ACL policy combo box, as shown in **Figure 7**, will result in a new look once you finish with the installer.

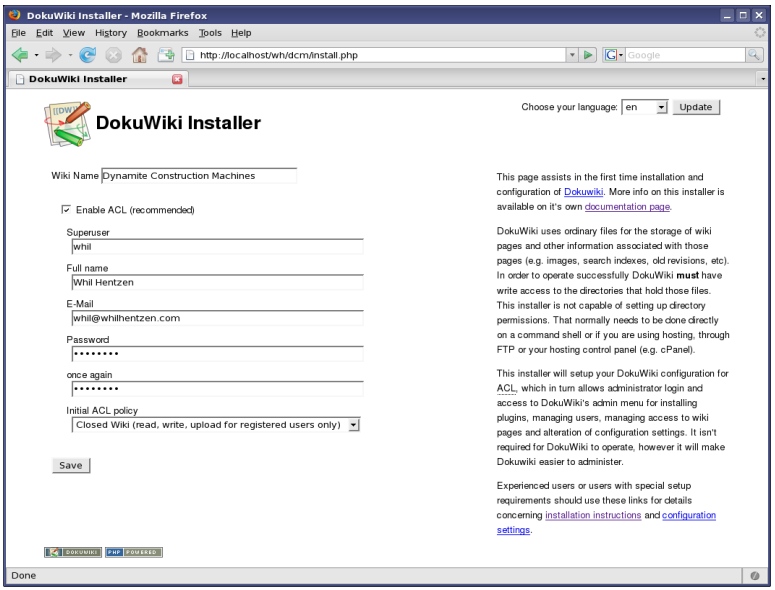

***Figure 7**. Selecting Closed Wiki in the DokuWiki Installer.*

As you can see in **Figure 8**, the initial page for a Closed wiki results in a reminder to login.

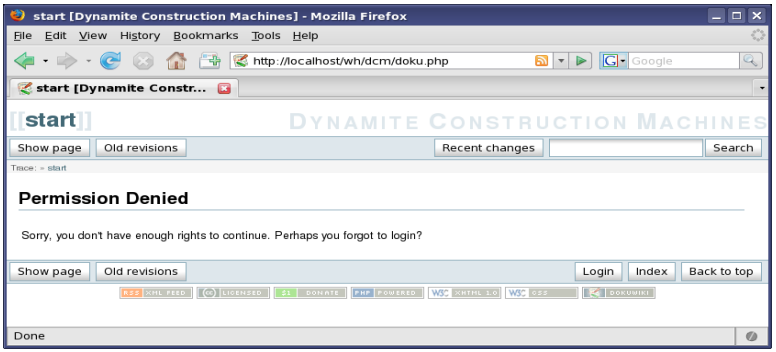

***Figure 8**. The home page for a Closed wiki requires login.*

All visitors must click "Login" and then enter their username and password credentials to gain access to the wiki - even if all they want to do is read pages. (There's a way to force the login page to automatically display instead of the possibly confusing "Permission Denied" page shown in Figure 8. See Chapter 15. *\\\ why is this a bad hack?)

Underneath the hood, the contents of the acl.auth.php file have changed.

```
*               @ALL            0
*               @user           8
```

These strings mean that all must log in before reading, editing or creating content (0).

The local.php and users.auth.php files remain the same.

From our brief exposure to ACL files, it appears that each line contains two components. The first is the name of the group and the second is a number that controls the permissions of the members of that group. In reality, things are somewhat more complex; I'll cover ACLs in more detail in Chapter 12.

Comparison of options

What if you chose one configuration and then realized you really wanted a different one? Since text files contain all of the configuration options, you can switch between ACL permutations by editing the acl.auth.php file. Here's a table that describes the basic differences.

Type	acl.auth.php Contents	Meaning
Open	`*  @ALL  8`	All can read and edit

Public	* *	@ALL @user	1 8	All can read Must log in to edit
Closed	* *	@ALL @user	0 8	Must log in to read or edit

After you've made the desired changes, you'll likely need to refresh the page. If you've really munged things up, you can just delete the local.php, users.auth.php and acl.auth.php files and go back to the DokuWiki installer:

```
http://localhost/<path_to_DokuWiki_root>/install.php
```

(You may have deleted install.php as suggested after you completed installation. In this case, you can re-upload it from your local copy.) Note that if you try to cheat and just hit your back button until you get to the Installer page (Figures 1, 3, 5 and 7), make a change, and then continue forward again as if no one was looking - without deleting the files - you'll get an error courtesy of a page that looks like **Figure 9.**

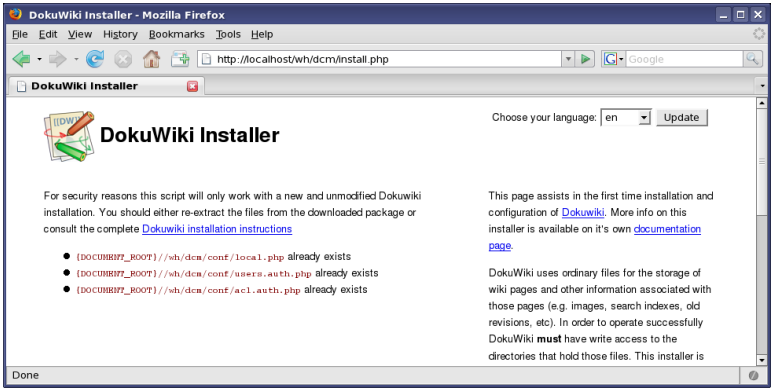

***Figure 9**. Rerunning the DokuWiki installer without deleting the core config files generates an error page.*

Greeted with the page in Figure 9, you can hit your Back button to get to the Installer page again, ***now*** delete the files, and then continue on.

Saving your settings

By the way, if you've already done a bunch of configuration or added a bunch of users, your local.php file will be chock full of custom settings, and there'll be a lot of data in your users.auth.php file. Should you try to rerun the Installer, it will detect the configuration file with settings in it and complain. The solution is to rename those files temporarily and let the Installer create new empty ones for you. Then set those empty files aside and name your own copies back to their original names. (Technically speaking, you shouldn't ever have to rerun the Installer.)

Cleaning up

Once you're done with installation to your satisfaction, it's good practice to delete the install.php file. You've got the original in your dw_masters folder, right?

Conclusion

Restricting access to your wiki is a critical step in setting it up. The wiki contains, at the least, valuable information that isn't appropriate to be released into the wild, and, very possibly, trade secrets and other confidential information. Thus, controlling who gets in is a process that you need to address early on, and become proficient at it. Using and configuring ACLs provide this capability.

Updates and corrections to this chapter can be found on Hentzenwerke's Web site, www.hentzenwerke.com. Click "Catalog" and navigate to the page for this book.